Stephen A. Wight
Klarquist Sparkman, LLP
121 S.W. Salmon Street, Suite 1600
Portland, Oregon 97204
(503) 226-7391

Figure 1

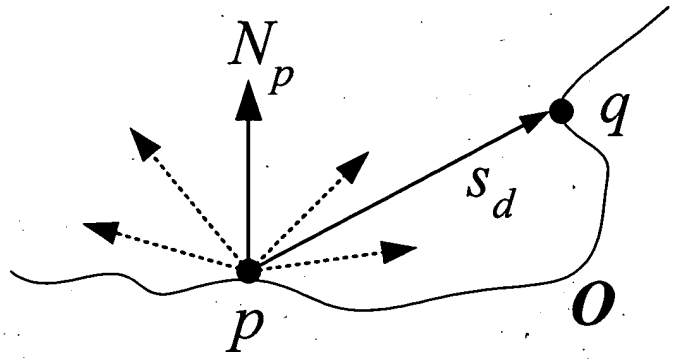
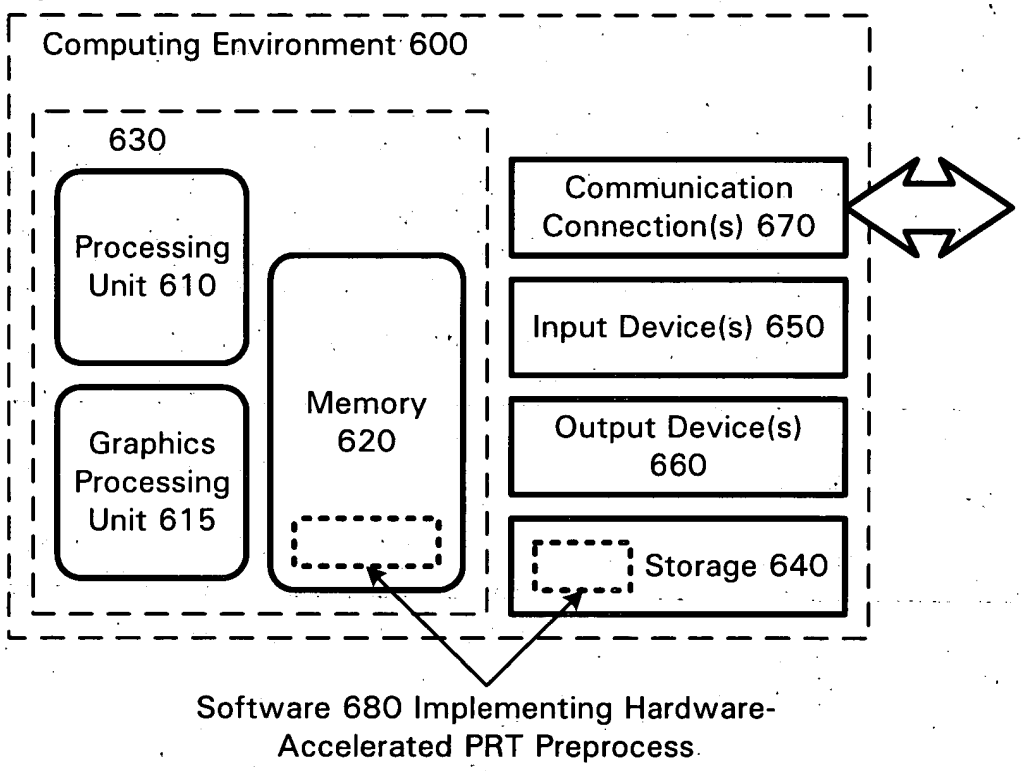
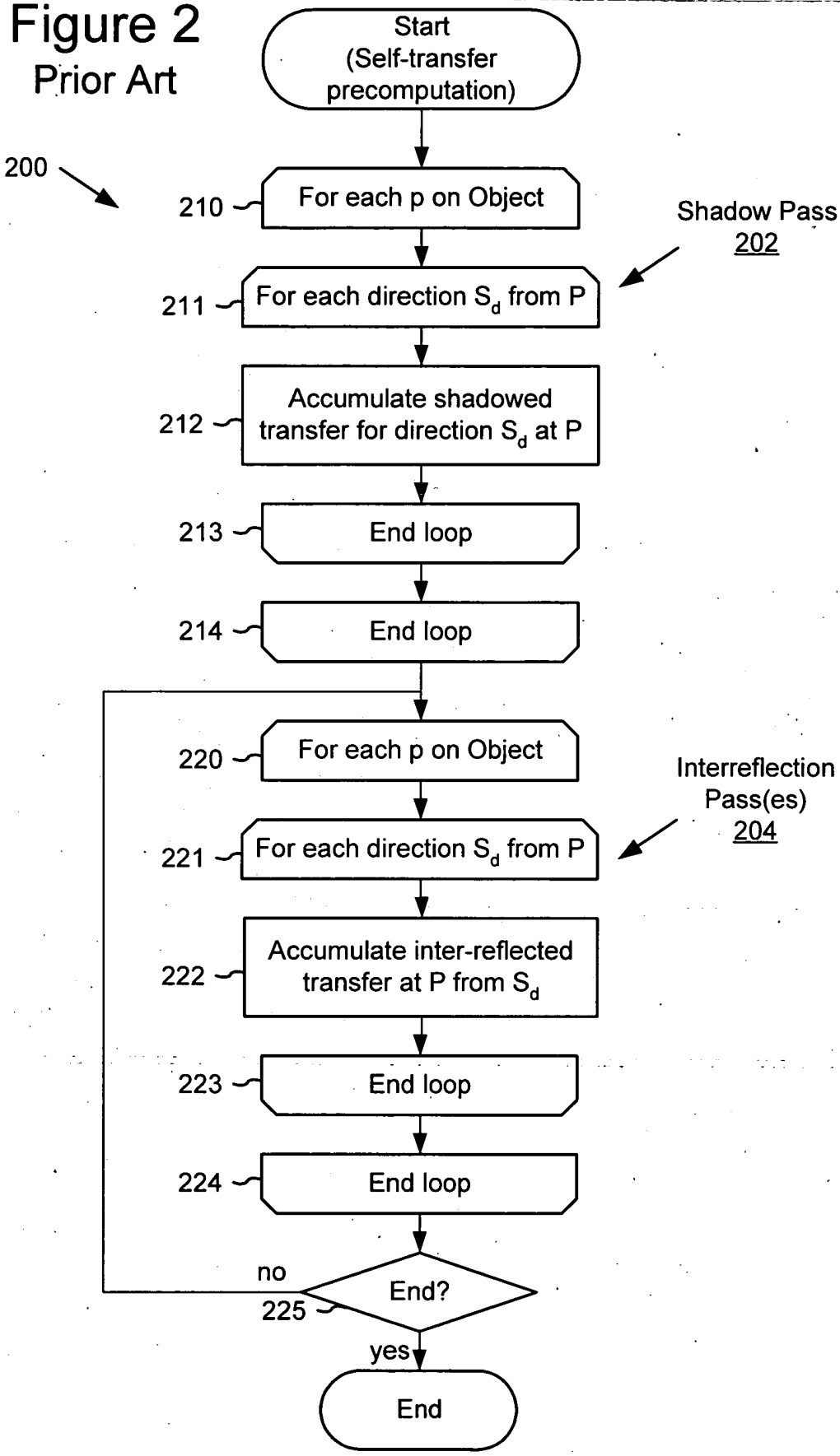


Figure 9



Stephen A. Wight
Klarquist Sparkman, LLP
121 S.W. Salmon Street, Suite 1600
Portland, Oregon 97204
(503) 226-7391

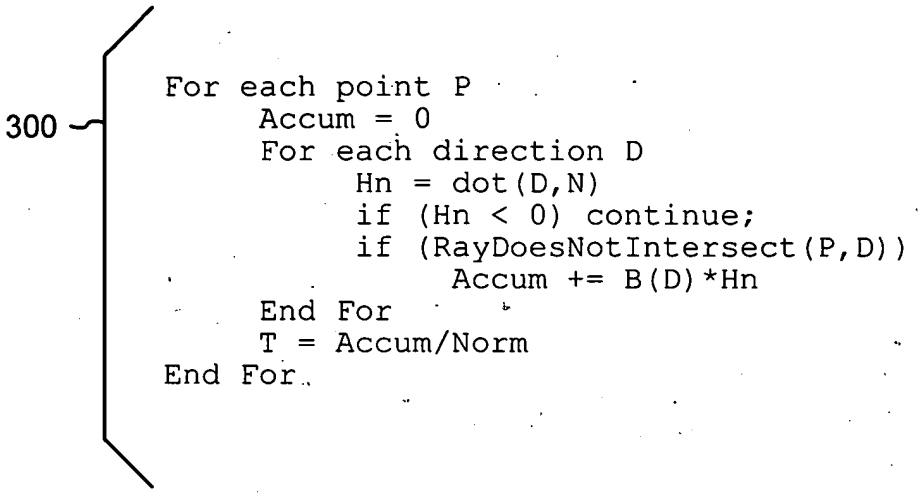
Figure 2
Prior Art



Stephen A. Wight
Klarquist Sparkman, LLP
121 S.W. Salmon Street, Suite 1600
Portland, Oregon 97204
(503) 226-7391

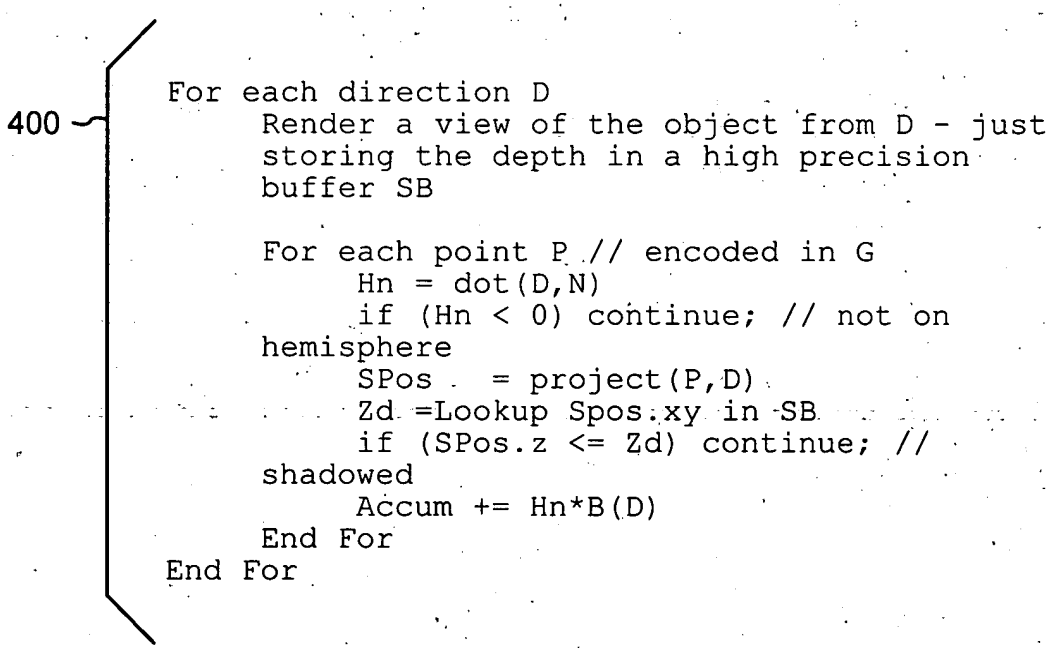
Figure 3

Prior Art



```
For each point P
     Accum = 0
     For each direction D
          Hn = dot(D,N)
          if (Hn < 0) continue;
          if (RayDoesNotIntersect(P,D))
               Accum += B(D)*Hn
     End For
     T = Accum/Norm
End For
```


Figure 4

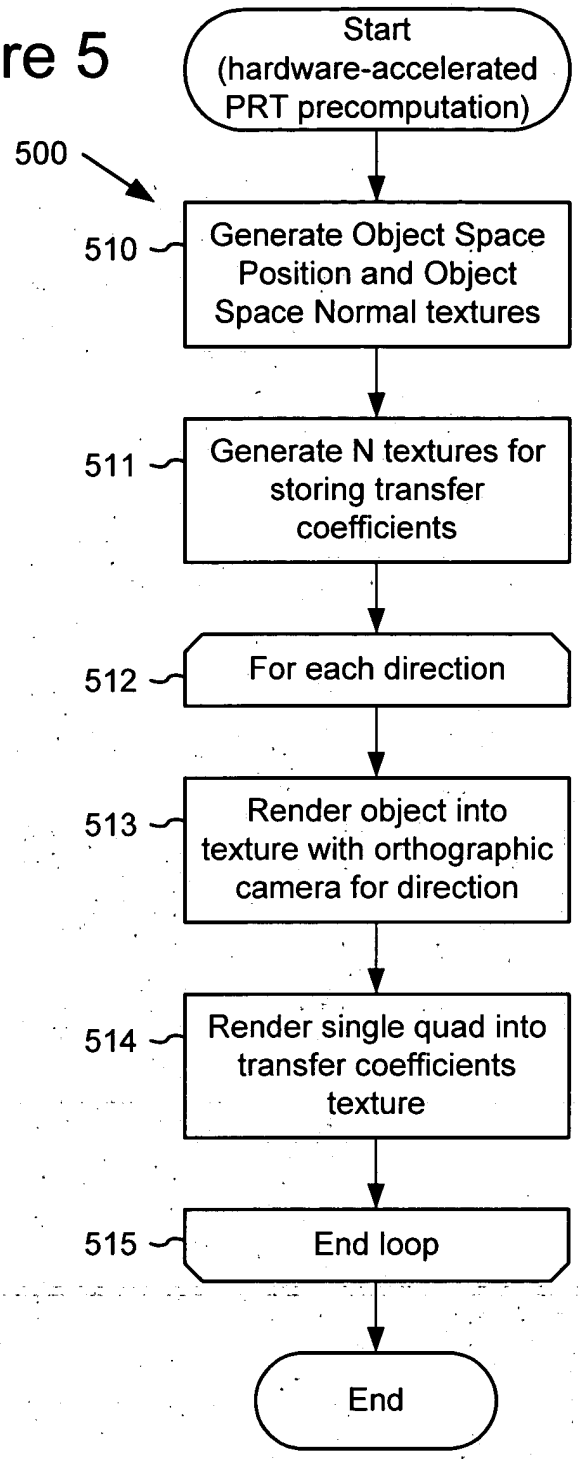


```
For each direction D
     Render a view of the object from D - just
     storing the depth in a high precision
     buffer SB

     For each point P // encoded in G
          Hn = dot(D,N)
          if (Hn < 0) continue; // not on
     hemisphere
          SPos  = project(P,D)
          Zd =Lookup Spos.xy in SB
          if (SPos.z <= Zd) continue; //
     shadowed
          Accum += Hn*B(D)
     End For
End For
```

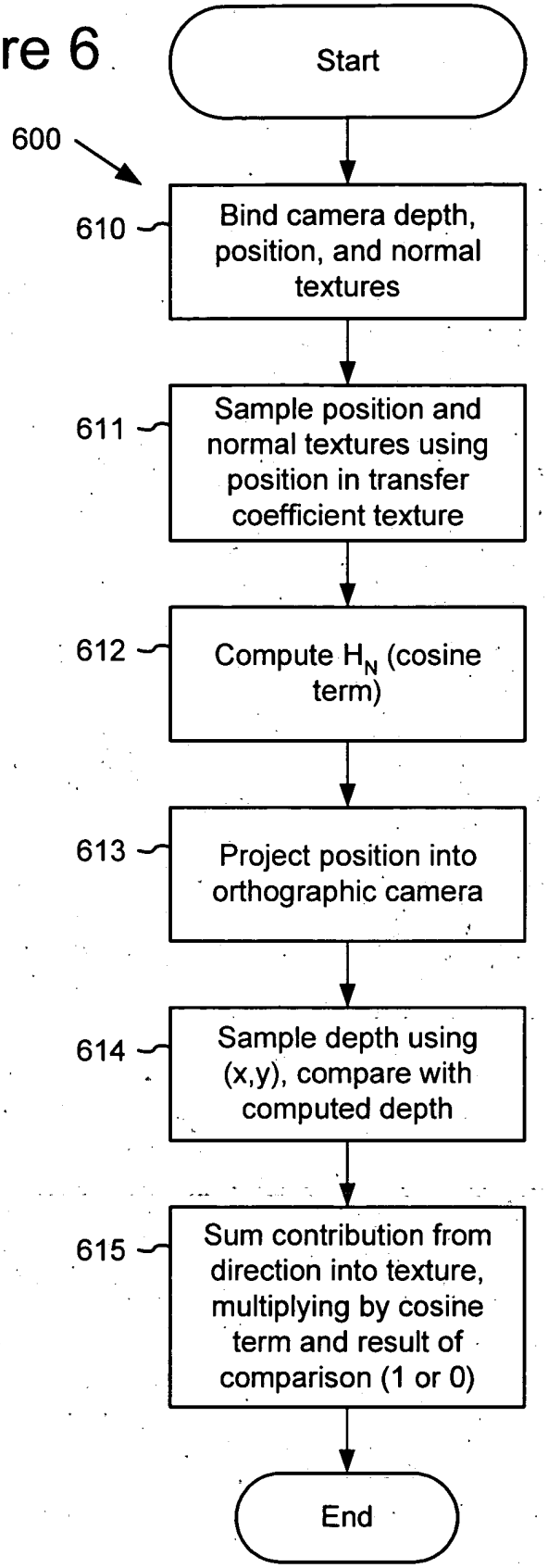
Stephen A. Wight
Klarquist Sparkman, LLP
121 S.W. Salmon Street, Suite 1600
Portland, Oregon 97204
(503) 226-7391

Figure 5



Stephen A. Wight
Klarquist Sparkman, LLP
121 S.W. Salmon Street, Suite 1600
Portland, Oregon 97204
(503) 226-7391

Figure 6



Stephen A. Wight
Klarquist Sparkman, LLP
121 S.W. Salmon Street, Suite 1600
Portland, Oregon 97204
(503) 226-7391

Figure 7

700

```
ps.2.0
;---------------------------
; Constants specified by the app
;
; c0-2  transformation into shadzbuff coordinates
;       xy values are mapped directly to texture
coordinates
; c3    direction vector for lighting
; c4-c7 sh coefficients for direction
; c8    is "default" texture coordinates - (0,0)

def c8, 0.0, 0.0, 0.0, 0.0

dcl t0;    // uv coordinates

dcl_2d s0; // geometry texture
dcl_2d s1; // normal texture
dcl_2d s2; // shadow zbuffer

// 1st sample the geom/normal from the source textures

texld r3, t0, s0;
texld r4, t0, s1; // should have a homogenous 1...

// normalizes the normal - not strictly neccesary
dp3 r0, r4, r4
rcp r0.x, r0.x
mul r0, r0.x, r4

// r0 is now the normalized normal
// compute dot product of normal with direction
dp3 r0, r0, c3

// r1 is going to be the transformed point
dp4 r1.x, r3, c0
dp4 r1.y, r3, c1
dp4 r1.z, r3, c2

// fudge texture coordinates if dot(N,V) < 0
// makes it more cache coherent
cmp r1.xy, r0, r1, c8

// sample the shadow buffer
texld r2, r1, s2

// do the subtraction
add r2, -r1.z, r2.x

// compute the clamped cosine value
cmp r0, r0, r0, c8
// compute the "multiplication" value - cos or zero
cmp r0, r2, r0, c8

// now move the 16 coefficients into the MRT
mul r1, r0, c4
mov oC0, r1
mul r1, r0, c5
mov oC1, r1
mul r1, r0, c6
mov oC2, r1
mul r1, r0, c7
mov oC3, r1
```

Stephen A. Wight
Klarquist Sparkman, LLP
121 S.W. Salmon Street, Suite 1600
Portland, Oregon 97204
(503) 226-7391

Figure 8

800

```
ps.2.0
;---------------------------
; Constants specified by the app
;
; c0     vectors of 1's
; c1-c4  red   lighting coefs
; c5-c8  green lighting coefs
; c9-c12 blue  lighting coefs

def c0, 1.0, 1.0, 1.0, 1.0

dcl t0;    // texture coords

dcl_2d s0; // 1st set of coefs
dcl_2d s1; // 2nd set of coefs
dcl_2d s2; // 3rd set of coefs
dcl_2d s3; // 4th set of coefs

// load the transfer coefficints
texld r0, t0, s0
texld r1, t0, s1
texld r2, t0, s2
texld r3, t0, s3

// accumulate 4 parts of the red dot product
mul r4, r0, c1
mad r4, r1, c2, r4
mad r4, r2, c3, r4
mad r4, r3, c4, r4
// sum across the 4 parts
dp4 r5.r, r4, c0

// accumulate 4 parts of the green dot product
mul r4, r0, c5
mad r4, r1, c6, r4
mad r4, r2, c7, r4
mad r4, r3, c8, r4
// sum across the 4 parts
dp4 r5.g, r4, c0

// accumulate 4 parts of the blue dot product
mul r4, r0, c9
mad r4, r1, c10, r4
mad r4, r2, c11, r4
mad r4, r3, c12, r4
// sum across the 4 parts
dp4 r5.b, r4, c0

// shove something in the alpha channel
mov r5.a, c0.r

//output the color

mov oC0, r5
```